Non-Responsive Amend. last paragraph + fourth. 6 months to buy. Defective O.A.R. wrong claims wrong reference.

#10 LDJ 2-28-03

Docket No. AT9-99-319 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Broussard** §

Serial No. **09/392,841** §

Filed: **September 9, 1999** §

For: **Method and System for Remote Java Audio Server in a Heterogeneous Distributed Environment** §

Group Art Unit: **2155**

Examiner: **Qureshi, Shabana**

Assistant Commissioner for Patents
Washington, D.C. 20231

ATTENTION: Board of Patent Appeals and Interferences

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on February 19, 2003.

By: Dell Whitton
Dell Whitton

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on December 19, 2002.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party: IBM Corporation

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-16

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: NONE
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1-16
4. Claims allowed: NONE
5. Claims rejected: 1-16

C. CLAIMS ON APPEAL

The claims on appeal are: 1-16

STATUS OF AMENDMENTS

There are no amendments after final rejection.

SUMMARY OF INVENTION

A method and system for an audio server in a heterogeneous distributed environment is provided. A Java application executes on a host machine under X Windows or RAWT (Remote Abstract Window Toolkit), and the Java application generates audio data and graphic data. The graphic data is sent to a display server on a client machine specified by a display environment variable. Although neither X Windows nor RAWT have audio support, a Java audio driver on the host machine determines whether an audio environment variable or an audio command line parameter is specified on the host machine. In parallel to the graphic data, the audio data is then sent to a Java audio server on the client machine specified by the audio environment variable or the audio command line parameter, and played using the local audio support, in Java, on the client machine on which the user can hear the audio.

ISSUES

The issues on appeal are as follows:

Whether claims 1-4, 6-10, 12, 13, 15-18, 20-24, 26, and 27 are unpatentable as being obvious over Traversat et al. (US Patent No. 6,119,157) in view of Subramaniam et al. (US Patent No. 6,070,187).

GROUPING OF CLAIMS

The claims on appeal do not stand or fall in a single group, but are grouped into in the following groups for the reasons set forth in the Argument section below:

Claims 1-3 and 14-16 form group A. Claims 4-13 form group B.

ARGUMENT

The Final Office Action rejects claims 1-16 under 35 U.S.C. § 103(a) as being unpatentable over *Trueblood* (U.S. Patent No. 5,748,499) in view of *Wong et al.* (US Patent No. 6,216,152). This rejection is respectfully traversed.

I. The Prior Art Fails to Teach or Suggest Sending Audio Data to a Platform-Independent Audio Server on a Client Machine (Group A)

With respect to claims 1 and 14, the Advisory Action, which provides a modified statement of the rejection, states:

> Trueblood teaches a method for a distributed audio server (column 2, lines 43-49), the method comprising the computer implemented steps of: generating audio data and graphic data (columns 5-6; column 21, lines 49-55); sending the graphic data to a display server on a client machine specified by a display environment variable (columns 5-7; 'X-commands'); and sending the audio data to an audio server on the client machine (column 7, line 60 – column 8, line 20) specified by an audio environment variable or by an audio command line parameter (column 6, lines 50-65; column 7; column 7, line 41 – column 11, line 34, 'X-command file', 'X-command'). Trueblood fails to teach that the application is overall platform independent because the audio server is communicates with hardware. Wong et al. teach a media plug-in application that is platform independent (abstract; column 5, lines 30-51). It would be obvious to one of ordinary skill in the art at the time the invention was made to employ Wong's teachings within the system of Trueblood because employing a platform independent application would allow the system to be hardware-tolerant, and therefore be able to run on various types of machines.

Advisory Action, dated November 13, 2002. Appellant respectfully disagrees. *Trueblood* teaches a "computer graphics data recording and playback system." See Title. During recording, the client executes an application on a server using the X-Window protocol. As stated by *Trueblood*, the X-server receives X-Window commands and events and converts them into device specific actions understandable by a graphic controller. There is no hint or suggestion whatsoever in *Trueblood* that an X-Window command stream includes audio data. This is a disadvantage associated with X-Window, which is recognized and solved by the present invention, but casually accepted by the reference. *Trueblood* makes no attempt to include audio data in an X-Window command stream.

Trueblood does teach recording X-Window commands and user interaction information, such as keyboard actions and mouse pointer movement. This recording function is performed at the client workstation; therefore, recording and playback is completely unrelated to the distributed audio server of the present invention. In addition, *Trueblood* teaches supplementing the X-Window recording with a time stamped audio track. *Trueblood* states:

> The system also provides for the addition of a time stamped audio track to be included with the playback. Such an audio track may be used, for example, to add the voice of an instructor for training purposes. Alternatively, it may be used to add a recording of radio communications which occurred during the actual events recorded.

Trueblood, col. 2, lines 43-49. As such, *Trueblood* teaches the addition of an audio track; however, whether the audio track is generated at the time the X-Window information is recorded is purely circumstantial. For example, in the example of audio for the voice of an instructor, this audio may be prepared before or well after the X-Window information is recorded. This audio data is also stored in association with the client

In contradistinction, the present invention provides an audio server that generates audio data and graphic data in a platform-independent application at the server. Claim 1 of the present application recites:

> 1. A method for a distributed audio server, the method comprising the computer-implemented steps of:
>
> generating audio data and graphic data in a platform-independent application;
>
> sending the graphic data to a display server on a client machine specified by a display environment variable; and
>
> sending the audio data to a platform-independent audio server on the client machine specified by an audio environment variable or by an audio command line parameter.

Claim 14 recites similar limitations for a computer program product. *Trueblood* fails to teach or fairly suggest "generating audio data and graphic data **in a platform-independent application**," as recited in representative claim 1. In *Trueblood*, the audio is apparently recorded from an outside source in a separate process from the recording of X-Window data.

Furthermore, Trueblood fails to teach or fairly suggest "sending the audio data to a platform-independent audio server on the client machine specified by an audio environment variable or by an audio command line parameter," as recited in representative claim 1. In fact,

Trueblood states:

> In the present invention, the graphics software is platform-independent and can operate on any work station embodying an appropriate operating system. However, when audio is included, the audio client communicates with the audio hardware of the work station, thus making the overall system no longer platform-independent.

Trueblood, col. 8, lines 6-11. Furthermore, the X-Windows protocol used in the *Trueblood* system cannot support platform-independent audio data streams, which is why *Trueblood* must rely on the platform-dependent audio playback method described above. This limitation of the X-Windows system is specifically pointed out in the present application as a problem to be solved by the present invention. See specification, page 2, line 27, to page 3, line15.

No portion of *Trueblood* teaches or suggests a method to overcome this limitation of the X-Windows protocol. Rather, *Trueblood* seems to accept this limitation and supplement the audio-less X-Window information with audio data from an outside source. By contrast, the invention recited in claims 1 and 14 does solve these disadvantages of the X-Windows protocol.

With regard to platform independence, the Advisory Action states:

> Trueblood fails to teach that the application is overall platform independent because the audio server is communicates with hardware. Wong et al. teach a media plug-in application that is platform independent (abstract; column 5, lines 30-51). It would be obvious to one of ordinary skill in the art at the time the invention was made to employ Wong's teachings within the system of Trueblood because employing a platform independent application would allow the system to be hardware-tolerant, and therefore be able to run on various types of machines.

Advisory Action, dated November 13, 2002. Appellant respectfully disagrees. The simple fact that the plug-in decoders taught in *Wong* are platform-independent plug-ins does not mean that they can be combined with *Trueblood* to produce the limitations of the present invention. In determining obviousness, an applicant's teachings may not be read into the prior art. *Panduit Corp. v. Denison Mfg. Co.*, 810 F.2d 1561, 1575 n. 29, 1 U.S.P.Q. 1593, 1602 n. 29 (Fed. Cir. 1987) (citing need to "guard against hindsight and the temptation to read the inventor's teachings into the prior art").

The plug-ins taught by *Wong* are designed to be used with applications, most notably web browsers. However, there is no teaching in either *Trueblood* or Wong of "generating audio data and graphic data in a platform-independent application" and "sending the audio data to a

platform-independent audio server on the client machine specified by an audio environment variable or by an audio command line parameter," as recited in representative claim 1.

Furthermore, the mere addition of plug-ins is not sufficient to overcome the inability of the *Trueblood* system to support platform-independent audio data streams, and it certainly does not produce a system that can support platform-independent audio data by means of a platform-independent audio server on a client machine. Therefore, the proposed combination of *Trueblood* and *Wong* does not produce the limitations of claims 1 and 14.

Still further, the Advisory Action states:

> In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin,* 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). The motivation to combine the feature of platform independent of Wong into the teachings of Trueblood is that it will allow the system of Trueblood to be used on a universal platform.

Advisory Action, dated November 13, 2002. Appellant respectfully disagrees. For the sake of argument, even assuming that *Trueblood* and *Wong* can be combined, the resulting combination still would not teach the limitations of the present invention for the reasons stated above. The present invention uses platform-independent audio data. *Trueblood* teaches an X-Windows system, which cannot support platform-independent audio data. *Trueblood* does not teach or suggest any method for overcoming this limitation of X-Windows. Rather, *Trueblood* explicitly accepts this limitation and makes no attempt to overcome it. As explained above, the plug-ins taught by *Wong* are not capable of overcoming this inherent limitation of X-Windows, nor is there any suggestion in *Wong* of how such plug-ins could be used to overcome this problem in X-Windows. Therefore, combining Wong with *Trueblood* (if technically feasible at all), still would not allow the *Trueblood* system to support platform-independent audio data.

Since claims 2, 3, 15, and 16 depend from claims 1 and 14, the same distinctions between *Trueblood* and *Wong* and the invention recited in claims 1 and 14 apply for these claims. Additionally, claims 2, 3, 15, and 16 recite other additional combinations of features not

suggested by the reference. Consequently, it is respectfully urged that the rejection of claims 1-3 and 14-16 is overcome.

II. The Prior Art Fails to Teach or Suggest Receiving Audio Data at an Audio Driver in a Server (Group B)

With respect to claims 4 and 9, the Final Office Action states:

> As per claims 4 and 9, Trueblood teaches a method for a distributed audio server (column 2, lines 43-49), the method comprising the computer implemented steps of generating audio data (abstract). Trueblood fails to teach that the audio server is implemented as a platform-independent application. Wong et al. teach a media plug-in application that is platform independent (abstract; column 5, lines 30-51). It would be obvious to one of ordinary skill in the art at the time the invention was made to employ Wong's teachings within the system of Trueblood because using a platform independent application would allow it to run on various types of machines.
>
> The method of determining whether an audio environment variable or an audio command line parameter is defined and if an audio environment variable or an audio command line is defined, sending the audio data to a platform-independent audio server on a client machine specified by the audio environment variable or by the audio command line parameter is inherent to Trueblood's invention (column 4, lines 55-67), the method is more further disclosed by Wong et al. (column 7, lines 1-37). Because Wong et al. refer to the method being applied to media, the method inherently includes audio data and graphic data. This provides the basis on which claims 5 and 10 are rejected. (Office Action, page 3)

Final Office Action, dated September 27, 2002. Appellant respectfully disagrees. This rejection is traversed for the same reasons explained above with regard to claims 1 and 14. Furthermore, the Appellant disputes the assertion that the use of an audio server and an audio environment variable are inherent to *Trueblood* and *Wong*. The Office Action misapplies the concept of "inherent" anticipation. Section 102 of Title 35 deals with novelty and loss of patent rights. An invention is said to be "anticipated" when it is squarely described or disclosed in a single reference as identified from one of the categories of 35 U.S.C. § 102, commonly referred to as "prior art." Express anticipation occurs when the invention is expressly disclosed in the prior art, patent or publication. In some cases, however, when the claimed invention is not described *in haec verba*, the "doctrine of inherency" is relied on to establish anticipation. Under the principles of inherency, a claim is anticipated if a structure in the prior art **necessarily** functions

in accordance with the limitations of a process or method claim. *In re King*, 801 F.2d 1324, 231 U.S.P.Q. 136 (Fed. Cir. 1986).

A prior art reference that discloses all of a patent's claim limitations anticipates that claim even though the reference does not expressly disclose the "inventive concept" or desirable property the patentee discovered. *Verdgaal Brothers, Inc. v. Union Oil Company of California*, 814 F.2d 628, 2 U.S.P.Q.2d 1051, (Fed. Cir. 1987). It suffices that the prior art process inherently possessed at that property. *Id.* Mere possibilities or even probabilities, however, are not enough to establish inherency. The missing claimed characteristics must be a "natural result" flowing from what is disclosed. *Continental Can Co. v. Monsant Co.*, 948 F.2d 1264, 20 U.S.P.Q.2d 1746 (Fed. Cir. 1991). Unstated elements in a reference are inherent when they exist as a "matter of scientific fact." *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 7 U.S.P.Q.2d 1057 (Fed. Cir.), *cert. denied*, 488 U.S. 892 (1988) and *Hughes Aircraft Co. v. United States*, 8 U.S.P.Q.2d 1580 (Ct. Cl. 1988). Otherwise, the invention is not inherently anticipated.

In the present case, the assertion that these elements are present can be made only through the use of the applicants' disclosure as a template to fill in the missing elements. As explained above, *Trueblood* teaches a system that operates using X-Windows, which does not support audio. Again, no part of *Trueblood* teaches or suggests a mechanism to overcome this limitation of X-Windows. Therefore, Trueblood cannot possibly teach, inherently or otherwise, "determining whether an audio environment variable or an audio command line parameter is defined," as recited in representative claim 4.

The Advisory Action states:

> In response to applicant's argument that an environment variable is not inherent to the system of Trueblood, X-Windows provides a display environment variable that allows a client to specify the display to which data or commands are being sent. It would be inherent to a system employing audio in X-Windows to have an audio environment variable that allows a client to specify the audio device to which data or commands are being sent.

Advisory Action, dated November 13, 2002. Appellant respectfully disagrees. As to "a system employing audio in X-Windows," there is no such system in the prior art. Therefore, what may or may not be inherent in such a system is irrelevant. The Advisory Action further states, "[w]ithout the audio environment variable, Trueblood's system would not be able to operate."

Appellant disagrees. *Trueblood* does not teach an audio environment variable and, presumably, the *Trueblood* system operates, albeit without audio.

As for *Wong*, the use of audio and video media does not inherently include the use of an audio environment variable and audio server or a display server, which are distinct from plug-in decoders. Moreover, as stated above, the plug-ins in *Wong* are used with applications such as web browsers, which are fundamentally distinct from the display and audio servers in the presently claimed invention.

Furthermore, the present invention provides a method in a distributed audio server that generates audio data in a platform-independent application and, in response to receiving the audio data at an audio driver, determines whether an audio environment variable or an audio command line parameter is defined. Claim 4 recites:

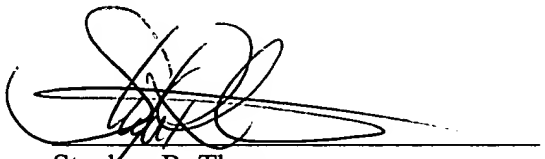
> 4. A method for a distributed audio server, the method comprising the computer-implemented steps of:
>
> generating audio data in a platform-independent application;
>
> in response to receiving the audio data at an audio driver, determining whether an audio environment variable or an audio command line parameter is defined; and
>
> if an audio environment variable or an audio command line parameter is defined, sending the audio data to a platform-independent audio server on a client machine specified by the audio environment variable or by the audio command line parameter.

Neither *Trueblood* nor *Wong* teaches or suggests receiving audio data, which is generated in a platform-independent application, at an audio driver, as recited in representative claim 4. The Final Office Action fails to address this feature altogether. In fact, neither *Trueblood* nor *Wong* mentions the term "audio driver" anywhere in their disclosures. The applied prior art, taken alone or in combination, fail to teach or fairly suggest each and every claim limitation. Therefore, claims 4 and 9 cannot be rendered obvious by the proposed combination of *Trueblood* and *Wong*.

Since claims 5-8 and 10-13 depend from claims 4 and 9, the same distinctions between *Trueblood* and *Wong* and the invention recited in claims 4 and 9 apply for these claims. Additionally, claims 5-8 and 10-13 recite other additional combinations of features not suggested by the reference. Consequently, it is respectfully urged that the rejection of claims 4-13 is overcome.

III. Conclusion

In view of the above, Appellant respectfully submits that the rejection of claims 1-16 under 35 U.S.C. § 103 are overcome. Accordingly, it is respectfully urged that the rejection of claims 1-16 not be sustained.



Stephen R. Tkacs
Reg. No. 46,430
Carstens, Yee & Cahoon, LLP
PO Box 802334
Dallas, TX 75380
(972) 367-2001

APPENDIX OF CLAIMS

The text of the claims involved in the appeal are:

1. A method for a distributed audio server, the method comprising the computer-implemented steps of:

generating audio data and graphic data in a platform-independent application;

sending the graphic data to a display server on a client machine specified by a display environment variable; and

sending the audio data to a platform-independent audio server on the client machine specified by an audio environment variable or by an audio command line parameter.

2. The method of claim 1 wherein the platform-independent application and the platform-independent audio server are implemented in the Java programming language.

3. The method of claim 1 wherein the display server is an X Windows display server.

4. A method for a distributed audio server, the method comprising the computer-implemented steps of:

generating audio data in a platform-independent application;

in response to receiving the audio data at an audio driver, determining whether an audio environment variable or an audio command line parameter is defined; and

if an audio environment variable or an audio command line parameter is defined, sending the audio data to a platform-independent audio server on a client machine specified by the audio environment variable or by the audio command line parameter.

5. The method of claim 4 further comprising:

generating graphic data in the platform-independent application; and

sending the graphic data to a display server on the client machine specified by a display environment variable.

6. The method of claim 4 wherein the platform-independent application and the platform-independent audio server are implemented in the Java programming language.

7. The method of claim 4 wherein the display server is an X Windows display server.

8. The method of claim 7 wherein the graphic data and the audio data are synchronized.

9. A data processing system for a distributed audio server, the data processing system comprising:

first generating means for generating audio data in a platform-independent application;

determining means for determining, in response to receiving the audio data at an audio driver, whether an audio environment variable or an audio command line parameter is defined; and

first sending means for sending, in response to a determination that an audio environment variable or an audio command line parameter is defined, the audio data to a platform-independent audio server on a client machine specified by the audio environment variable or by the command line parameter.

10. The data processing system of claim 9 further comprising:

second generating means for generating graphic data in the platform-independent application; and

second sending means for sending the graphic data to a display server on the client machine specified by a display environment variable.

11. The data processing system of claim 9 wherein the platform-independent application and the platform-independent audio server are implemented in the Java programming language.

12. The data processing system of claim 9 wherein the display server is an X Windows display server.

13. The data processing system of claim 12 wherein the graphic data and the audio data are synchronized.

14. A computer program product on a computer-readable medium for use in a data processing system for a distributed audio server, the computer program product comprising:

instructions for generating audio data and graphic data in a platform-independent application;

instructions for sending the graphic data to a display server on a client machine specified by a display environment variable; and

instructions sending the audio data to a platform-independent audio server on the client machine specified by an audio environment variable or by an audio command line parameter.

15. The computer program product of claim 14 wherein the platform-independent application and the platform-independent audio server are implemented in the Java programming language.

16. The computer program product of claim 14 wherein the display server is an X Windows display server.